UNIT 15 COMPUTER PROGRAMMING AND LANGUAGES

Structure

15.1 INTRODUCTION

As we know to solve a problem by computer we have to program the computers, means we have to write a program for solving the problem. Program is a set of instructions or statements written in any programming language. Programming language is the only mean by which we can communicate with computer. Broadly we can categorize programming language as low level language e.g. machine level, assembly level, and high level language (e.g. BASIC, COBOL, C, Visual Basic, JAVA etc.). Low-level languages are hard to learn and only be used by expert programmer while high-level programming languages can also be learned by non-computer professionals.

15.2 OBJECTIVES

After reading this unit, you should be able to :

- Explain the vocabulary used in programming languages;
- Identify and describe control structures in programming languages;
- Identify important features of Visual Basic, Java, HTML, Excel and COBOL; and
- Write programs in higher level language.

15.3 PROGRAMMING VOCABULARY

Suppose we want to write program for finding factorial of a number. The flow chart and pseudo code for this will be as follows:

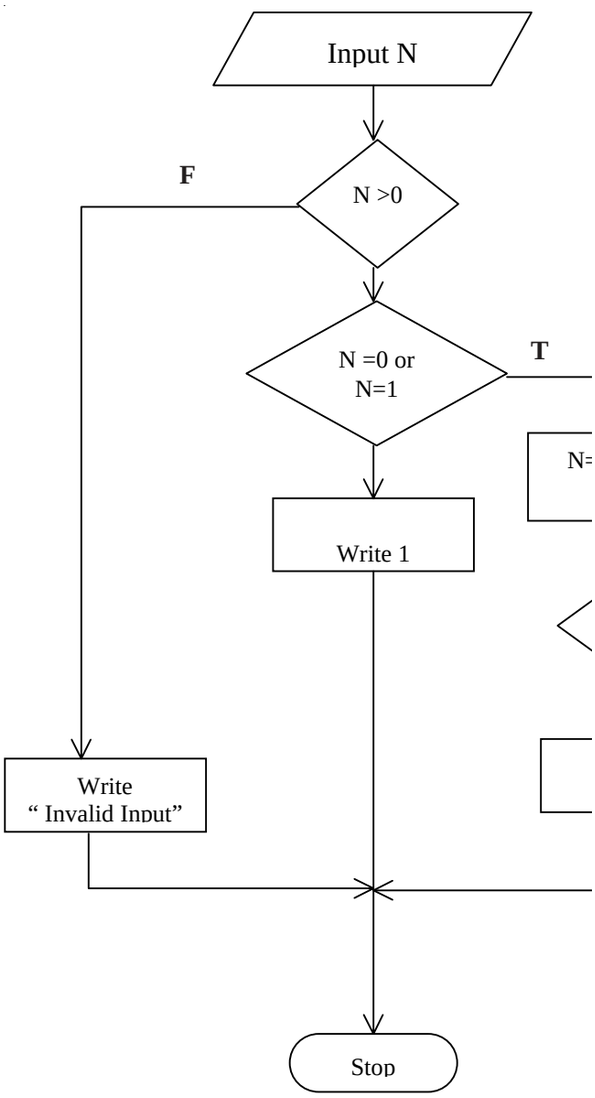


Fig. 15.1 (a)

```
Line  1  :   Input n
Line  2  :   if (n>0)
Line  3  :         if ((n=0) or (n=1))
Line  4  :               Write (1)
Line  5  :         else
Line  6  :                     Repeat
Line  7  :                           n=n*(n-1)
Line  8  :                           n=n-1
Line  9  :                     Until(n>0)
Line  10 :               Write n
Line  11 :   else
Line  12 :         Write “Invalid Input”
Line  13 :   end
```

Fig. 15.1 (b)

In writing above program we have used some important words such as if – then else, repeat-until etc. Meaning inherent in these words form fundamental concepts in programming. These words are called reserve words or keywords of a programming language. To write program in any language is simply a matter of learning its syntax and semantic rules and special features.

Apart from key words a program uses several other programming concept viz. identifiers, constants, expression, library functions etc.

Identifiers

Identifiers are the name given to various program elements such as variables, functions, sub-routines etc. Identifiers consist of letters and digits, in any order, except the first character must be a letter.

Variables

Variables are those elements of program whose value can change during the execution of a program.

Constant

Constant are those elements of program whose value cannot be changed during the execution of the program. Means they have fixed value.

The numeric constant can be integer representing whole number or floating point number with a decimal point. Constants would be probably the most familiar concept to us since we have used them in doing everything that has to do with numbers. Operations like, addition, subtraction, division, multiplication, and comparison can be performed on numeric constants.

String constants are sequence of alphanumeric characters enclosed in double quotation marks e.g. “Computer”, “X123”. String constant can be concatenated and compared in a lexicographic sense.

Expression

An expression represents a single data item, such as a number or a character. The expression may consist of a single entry. Such as constant, a variable, and array element or a reference to a function. It may also consist of some combination of such entities inter connected by one or more operators. Generally every high level programming language provides some sets of arithmetical, relational, logical operators. The most common arithmetical, relational, and logical operators are:

Arithmetical operator

*	Multiplication
/	Division operators
–	Subtraction operator
+	Addition operator

Relational operator

=	Equal to
<>	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Logical Operator

NOT Logical NOT operator simply negates a truth-value.

AND Logical AND operator return true if both operators are true (*e.g.* (4<=8) and (a<=b) is true.

OR Logical OR operator return false if both operands and false.

XOR Exclusive OR is true only if one of the operand is true and other is false.

Each operator has got some precedence associated with it and expression is always evaluated on the basis of precedence of the operator *e.g.*

3+7*6-4/2–2 ↑ 4

In above expression exponential operator is evaluated first then multiplication and division operators are evaluated and lastly the addition and subtraction operator are evaluated.

The precedence of arithmetic operators is as below:

Exponential operator	↑	(highest precedence)
Negation	–	
Multiplication an Division	* /	
Addition and subtraction	+ –	(Lowest precedence)

Relational operators have same level of precedence among themselves but low precedence than arithmetic operators. Logical operators have lower precedence than relation operators. Among themselves they have the following order of priority during evaluation of logical expression (logical expression is an expression evolving relational or logical operators and evaluate either true value or false value)

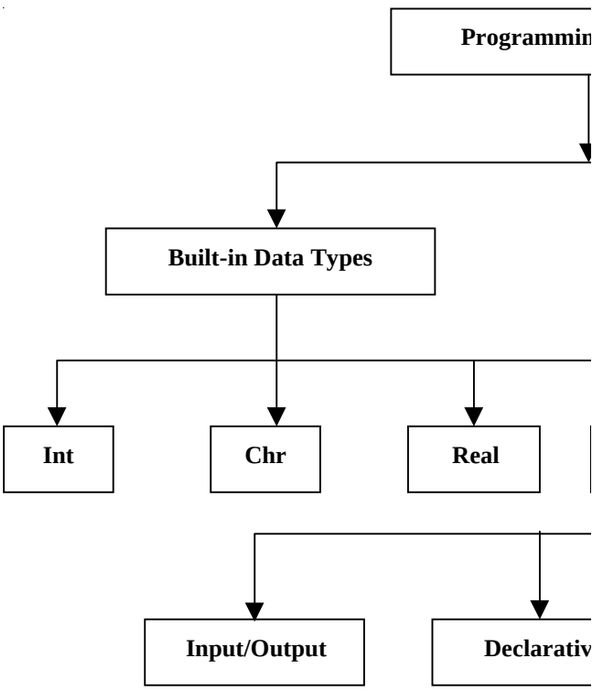
NOT	Highest precedence
AND	
OR, XOR	Lowest precedence

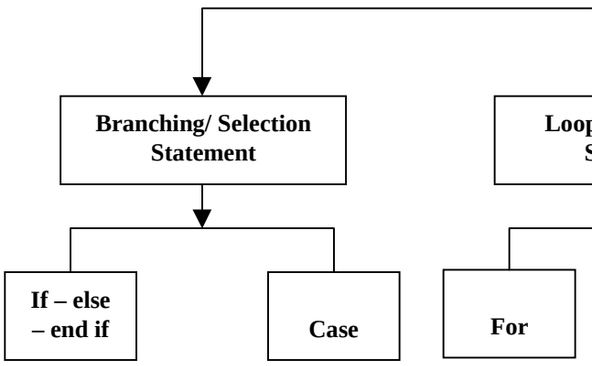
Expressions would be our basic workhorses. Our programs would be full of them, and while executing the program; computer would be preoccupied most of the time to evaluate them. But it would also be doing other things such as taking some values as input from input device, assigning a value or result of an expression to a variable, send a value for display to output device pondering to be or not to be on some conditional expression, repeating some tasks until we are satisfied and calling some slave – program to perform specific task. In the following discussion we shall elaborate on these actions.

Generally every high level programming language has got some:

i) Built in data types

ii) Statements


Programmin
Built-in Data Types
Int
Chr
Real
Input/Output
Declarativ


Branching/ Selection
Statement
Loop
S
If – else
– end if
Case
For

Built in data type and statement type are specific to a particular programming language.

Built in Data Types

Generally all programming language support following types of built in data *viz.*

Int.	Whole number
Real	Number with fractional part
Boolean	Data type which is either ‘true’ or ‘false’
Char	Single alpha number character enclosed in single quotation mark *e.g.* ‘a’, ‘1’ etc.
String	Group of characters enclosed in double quotation mark.

Before using any variable in the program it is generally necessary to define it’s type. The memory requirement for each data type is different and it is dependent upon the specific programming language.

Statements

Statement in programming language can be broadly classified as follows:

i) Declarative statements
ii) Input output statement
iii) Assignment statement
iv) Control statement or control structure.

Declarative Statements

These statements are used to declare the type of the variable e.g.

In Java

Int	x	:	is of integer type variable
Char	x	:	is character type variable
Byte	b	:	b is of byte type variable
Float	y	:	y is of float (real) type variable

In Visual Basic

Dim x as Integer	:	x is integer type variable
Dim b as Byte	:	b is byte type variable
Dim y as Single	:	y is of floating (real) type variable

Input Output Statement

They can be broadly classified as:

i) Console input statement
ii) File input statement
iii) Console output statement
iv) File output statement

Console input is used for taking the input from default input devices (*e.g.* line 1 of Figure 15.1(a)) while console output statement is used for printing the result of the program on default output device (e.g. line 4, line 10, line 12 of Figure 15.1(b)). Similarly file input statements are used for taking the input from file or non default input device while file output statement is used for writing the output into the specified file.

Assignment statements are used for assigning the value to variable e.g. variable name = value to be assigned.

X = 123 : value 123 is assigned to x and x is of integer type variable

Z = func1( ) : What so ever value return func1( ) will be assigned to variable z. The type of z and type of value returned by func1( ) should of same type.

15.4 CONTROL STATEMENT OR CONTROL STRUCTURE

The execution of computer program is always sequential *i.e*. statement which is written first will be executed first. To change the sequential execution of a program we need control structures, which can be broadly categorized as follows:

i) Branching or selection statement

ii) Looping or Iterative statement

iii) Jump statement

i) **Branching or Selection Statements** are used to select one group of statements or single statement for execution from several available groups of statements. The selection of a particular group depends on some Boolean condition.

If – then – else

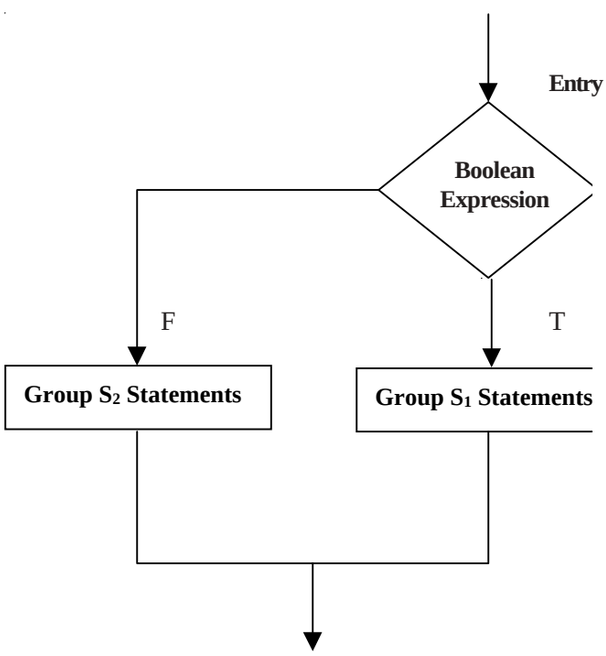
This statement used when we have to make selection from two available groups of statements. S1 group of statements will be only executed if Boolean expression true otherwise s2 group of statement will be executed.

If < Boolean expression >

S_1 statement (s)

Else

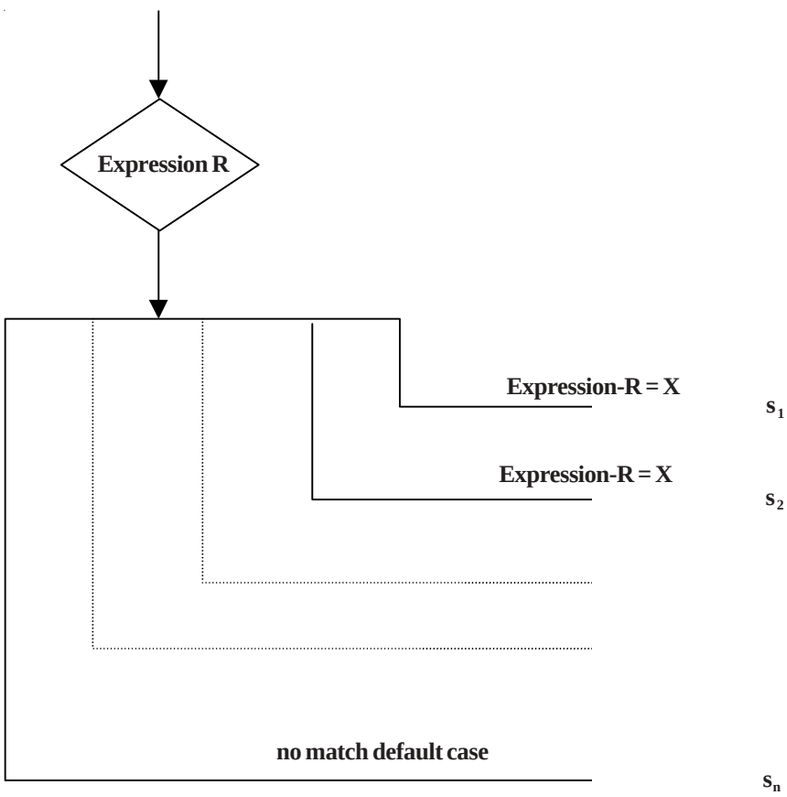
S_2 statement (s)



CASE statement is used if we have to make selection from several available group of statement

```
Case (expression R)
 {      Case  x        :      S1
        Case  y        :      S2
        Case  w        :      S3
        Case  z        :      S4
        ……..
        ……..
        default :      default statement
 }
```

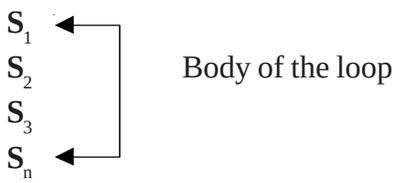
Here x, y. z ….w are permissible values for expression – R. If expression – R evaluates to y then S_2 group of statements will be executed if expression – R evaluates to non-permissible values then default group of statements will be executed.



ii) **Looping / Iterative Statement :** A computer is well suited to perform repetitive operations. It can do it tirelessly. Every computer language must have features that instruct a computer to perform such repetitive tasks. The process of repeatedly executing a block of statements is known as looping. The statements in the block may be executed any number of times, from zero to infinite number. If a loop continues forever it is called an infinite loop. Finite loop

is a loop, which terminates after fixed number of iterations. These fixed numbers of iterations are knows to us in advance for implementing finite looping we use for loop structure:

For Variable x = Value1 to Value 2 step values

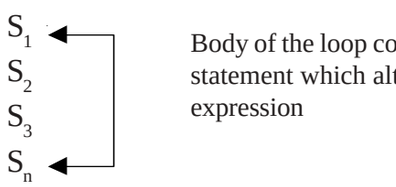


End

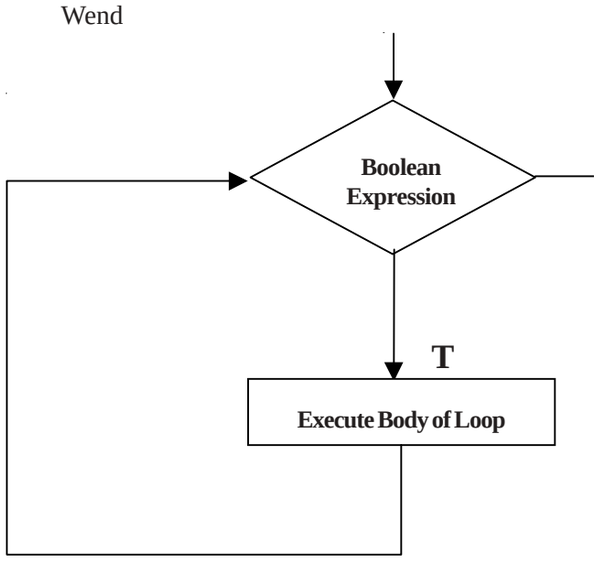
In this case variable is initialized to value1, then it is check whether the value of variable is greater value 2. If not then execute the body of the loop and then increase the value of variable by step size equal to value 3.

If we do not know in advance how many times the loop will iterate then we use while – loop or repeat – until structures. There types of loops are terminated by some Boolean expression. The body of these loops must contain one statement that alters the value of the Boolean expression.

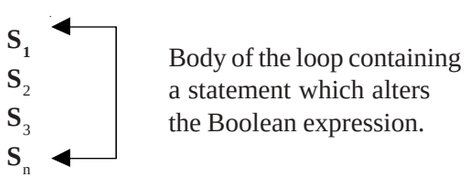
While < Boolean expression>



Wend



Repeat



Until <Boolean expression>

It is clear from the flow chart of while loop and repeat – until loop that body of repeat until loop will always executed at least for one time.

iii) **Jump Statement :** Go to, break and continue are generally the jump statements. Go to statement is used for jumping from one location to another labeled location within the program. Break statement causes the loop to be terminated, the continue statement causes the loop to be continued with the next iteration after skipping any statements in between.

Functions/ Sub-routine/Sub program/Modules

There are two ways to write a large and complicated computer program. One is to write the single program, if we can, which are normally very difficult for a large and complicated program. The other is to look at the large program as consisting of several somewhat less complicated subprograms. Coping with each of the subprogram can be seen in the same manner as main program, either write them directly, if we can, or refine it in terms of further subprograms and so on. The process of breaking large and complicated programs in smaller and less complicated sub program is known as modular programming approach. Each subprogram is termed as function, subroutine, module etc. depending on the specific programming language used to write the program.

Table 15.1: Languages Used in Different Applications

Application Type	Language used
1. General purpose programming	C, C++, Pascal, JAVA, Visual C++
2. GUI (Graphical User Interface)	Visual Basic
3. Database Management	Front end: Visual Basic Back end: Oracle, MS-Access, SQL Server
4. Artificial Intelligence	LISP, Prolog, Ada
5. Web Page Design/ Web enabled services	HTML, DHTML, XML, JAVA Applet, ASP etc.

15.5 OVERVIEW AND FEATURES OF VISUAL BASIC

The following points make Visual Basic an excellent development tool:

1. Visual basic applications are event driven. Even driven means the user is in control of application. The user generates a steam of events each time he or she clicks with the mouse or processes a key on the keyboard. Visual Basic application responds to those events through the code, you have written and attached to those events.

2. Visual Basic supports the principles of Object-oriented design. This means that you can compartmentalize different aspect of your application as objects and develop and test those objects independently of the rest of the application.

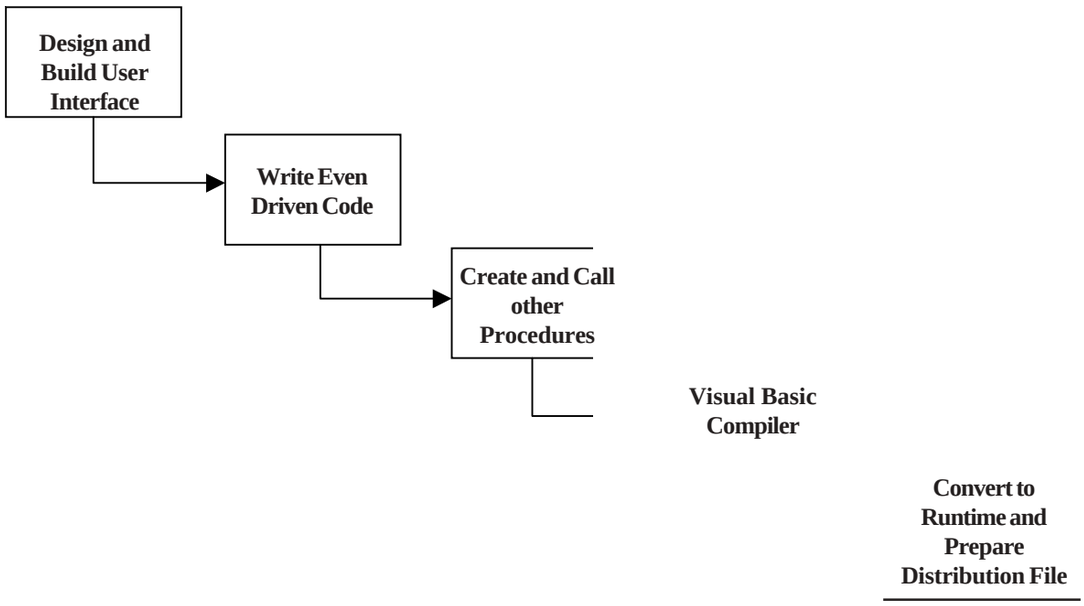
3. Microsoft has designed Visual Basic to be a complete windows application development system. This means that your visual basic applications will look and behave like other windows program.

4. Visual Basic is infinitely extensible through the use of Active X controls, dynamic ally linked libraries (DLLs) and add-ins. You can create these Active X controls, DLLs, and add-ins with Visual Basic 6 or buy them off the shelf from a large number of third party software vendors.

Visual basic program development overview

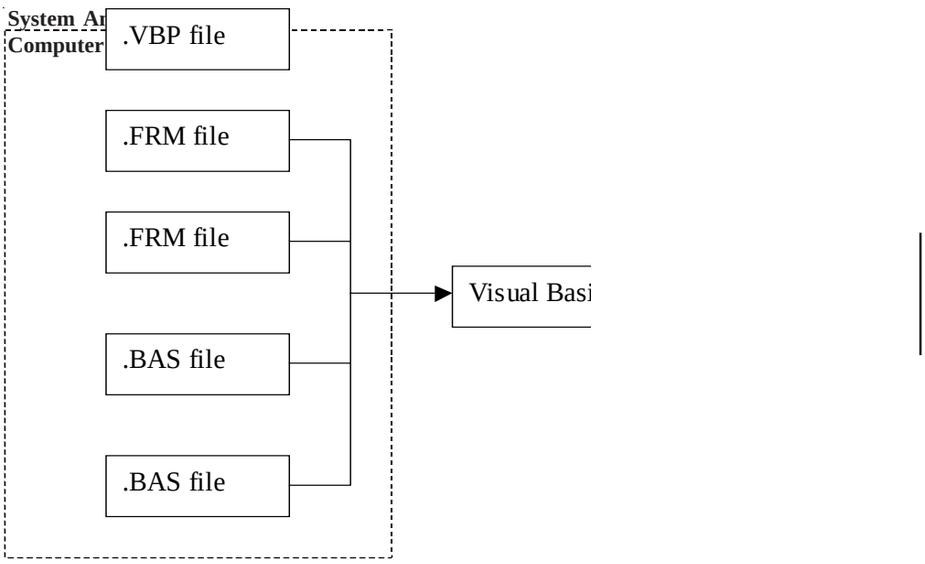
Visual Basic developers follow the following steps to develop a program:

1. Design and build the user interface
2. Write code that responds to events
3. Create and call other procedures and needed
4. Compile the program
5. Convert to runtime version
6. Prepare distributable set of files



As we work with Visual Basic to produce our windows application, we create several different files. An .FRM file contains the specifications and code associated with a single form within the application and a .BAS file contains only Visual Basic code. After we are satisfied with the design and operation of your Visual Basic application we can compile it into a distributable version. During the compilation process Visual Basic combines the information contained in the form file and code modules into a single executable file with an .EXE filename extension. The .FRM and .BAS files remain on the disk to enable up to continue making changes and improvements to the application. The figure below shows the schematic of the compilation process

As we work with Visual Basic to produce our windows application, we create several different files. An .FRM file contains the specifications and code associated with a single form within the application and a .BAS file contains only Visual Basic code. After we are satisfied with the design and operation of your Visual Basic application we can compile it into a distributable version. During the compilation process Visual Basic combines the information contained in the form file and code modules into a single executable file with an .EXE filename extension. The .FRM and .BAS files remain on the disk to enable up to continue making changes and improvements to the application. The Figure below shows the schematic of the compilation process



The .VBP file at the top of the diagram is the Visual Basic project file. This file contains references to all the components of the project and is not actually compiled into .EXE file.

A Visual Basic executable requires a large runtime module named MSVBM60.DLL. Usually MSBVM60.DLLis placed into the windows system directory as Visual Basic is installed.

Table 15.2: Build-In Data Types of Visual Basic

Data Type	Storage Requirement	Range
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,647
Single	4 bytes	For –ve values -3.402823E38 to –1.401298E-45 For + ve values 1.401298E-45 to 3.402823E38
Double	8 byte	For –ve values -1.79769313486232E308 to -4.94065645841247E-234 For +ve values 4.94065645841247E-234 to 1.79769313486232E308
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Boolean	2 bytes	True or false
Date	8 bytes	January 1,100 to December 31,9999
Object	4 bytes	Reference to any type of object
String	Varies	0 to Billion
Variant	Varies	Same range as any numeric or string dates type.

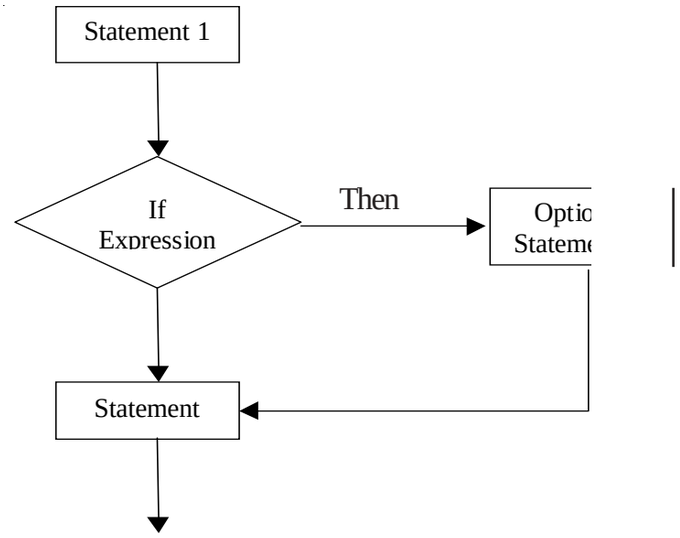
Control statement of Visual Basic

Following types of conditional branching statement is present in Visual Basic

i) If – then – end
ii) If – then – else
iii) If – then – Else if - End if

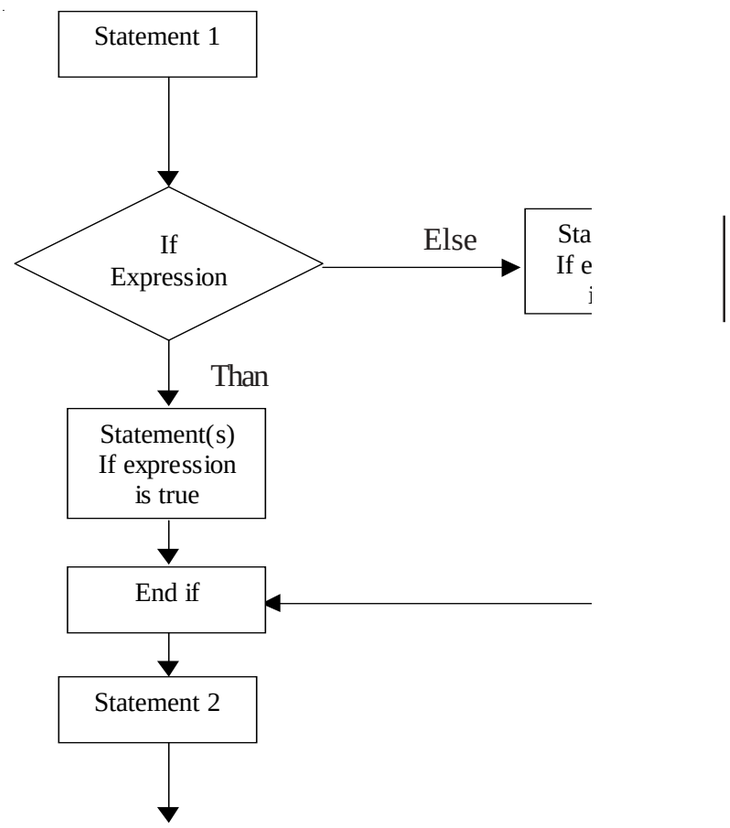
If- then is most common conditional branching statement. This is simplest and illustrated below. Syntax of If- then – end is

If *expression* then
Statement(s)
End if



The optional statement block is executed only when expression is true. Regardless of the value of the expression, execution continues after the if statement with statements.

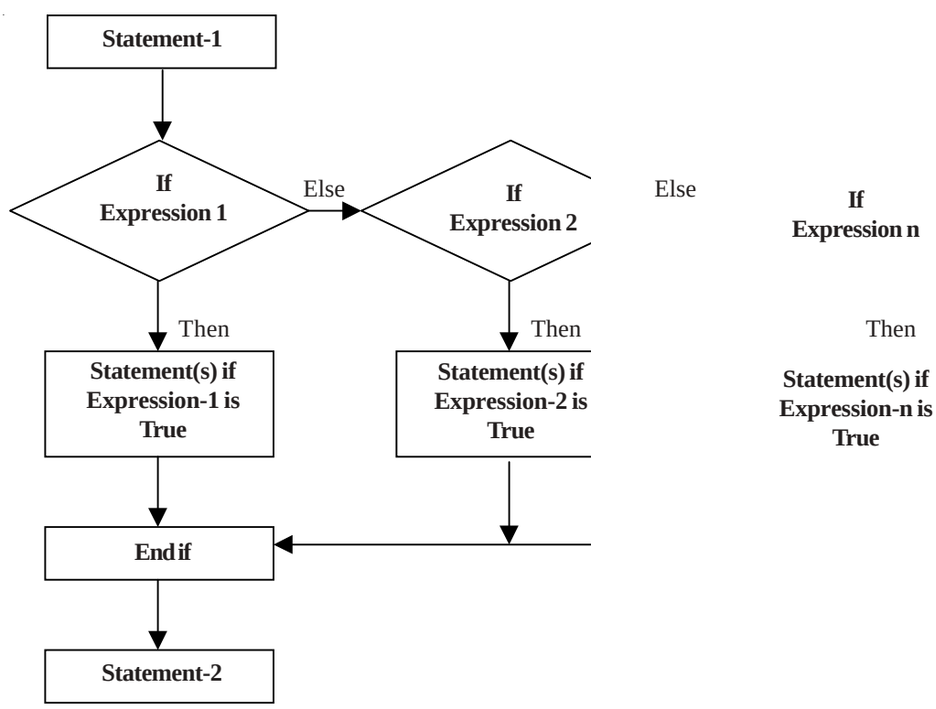
If then else is used when certain statements execute only when the expression is true other statements execute if the expression is false



Syntax:

If *Expression* Then
Statements(s)
Else
Statements(s)
End if

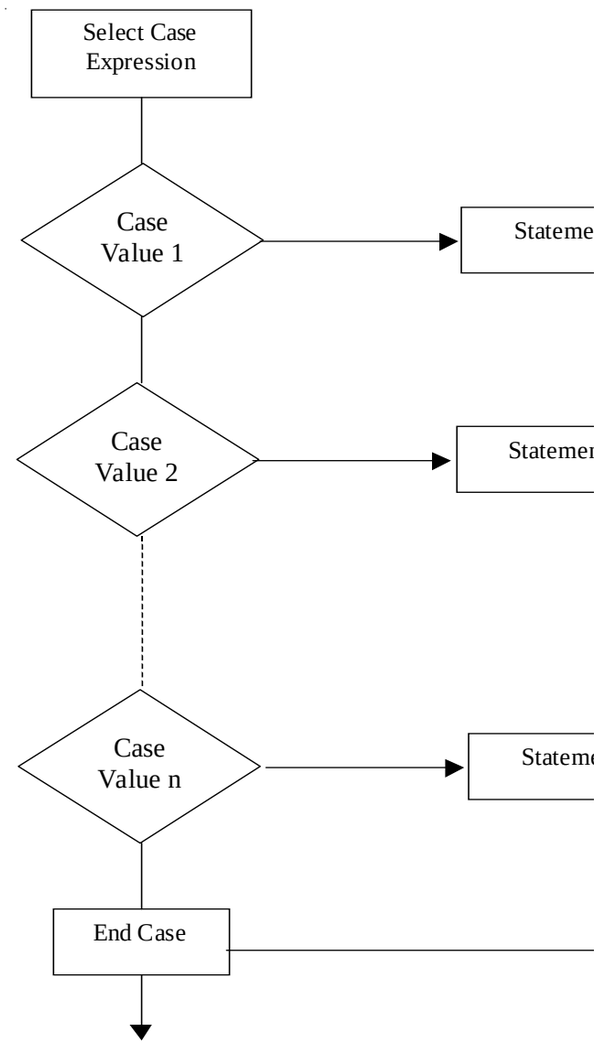
The final extension of if then End if branch is diagrammed below:



Syntax:

If *expression 1* then
Statement(s)
Else if *expression 2* then
Statement(s)
:
:
:
Else if *expression n* then
Statement(s)
End if

The select case branch is used in place of nested if then else statements and is easy to understand and is diagramed below:



Syntax:

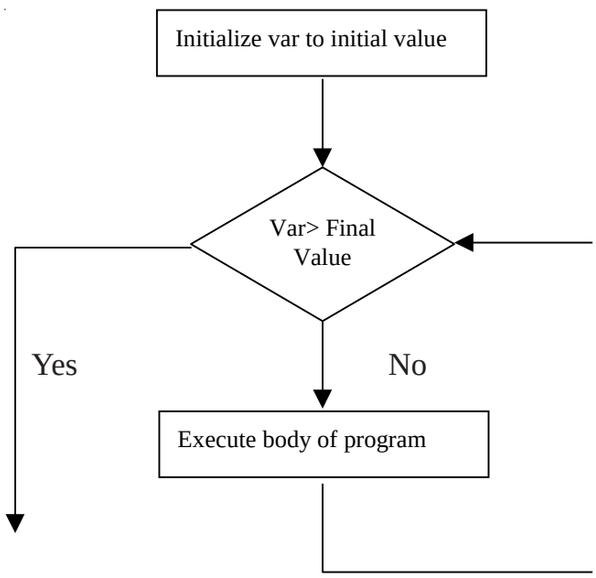
Select Case ***Expression***
    Case value 1
    Statement(s)
Case value 2
    Statement(s)
:
:
:
Case value n
    Statement(s)
Else Case
    Statement(s)
End Select

Looping Constructs of Visual Basic

The For… Next is useful whenever we know how many times loop through the statement block

Syntax

For *var* = initial value to final value
  Statement 1
  :
  :
  Statement n
Next *var*



Do – Loop executes a block of statements for as long as a condition is true. There are two variations of the Do – Loop statements, but both use the same basic model.

The loop can be executed either while condition is true or until the condition becomes true. The two variations of the Do – loop use the keywords while until to specify how long the statements are executed. To execute a block of statements while a condition is true, use the following syntax:

Do while ***condition***

Statement block

Loop

To execute a block of statements until the condition becomes true use the following syntax:

Do Until *Condition*

Statement block

Loop

15.6 OVERVIEW AND FEATURES OF JAVA

Java is a general purpose, object – oriented language developed by Sun Microsystems of USA in 1991. The most striking feature of the language is that it is platform – neutral language. Java is the first programming language that is not tied to any particular hardware or operating system. Programs developed in Java can be executed anywhere on any system. The important feature of Java is as under:

1. Compiled and Interpreted
2. Object Oriented
3. Distributed
4. Multi threaded

Usually, computer language is either compiled or interpreted. Java combines both these approaches thus making Java a two-stage system. Java compiler translates source code into what is known as byte code instruction. Byte codes are not machine instructions and therefore, Java interpreter generates machine code that can be directly executed by the machine that is running Java program.

Java is a true object oriented language. Almost everything in Java is an object. All program code and data reside within objects and classes.

Java is designed as a distributed language for creating applications on networks. It has the ability to share both data and programs. Java applications can open and access remote objects on Internet as easily as they can do in a local system. This enables multiple programmers at multiple locations to collaborate and work together on a single object.

Multithreaded means handling multiple tasks simultaneously. Java supports multithreaded programs. This means that we need not wait for the application to finish one task before beginning another e.g. we can listen to an audio clip while scrolling a page and at the same time downloaded an applet from a distant computer. Java does not support multiple inheritance and pointers.

Java Environment

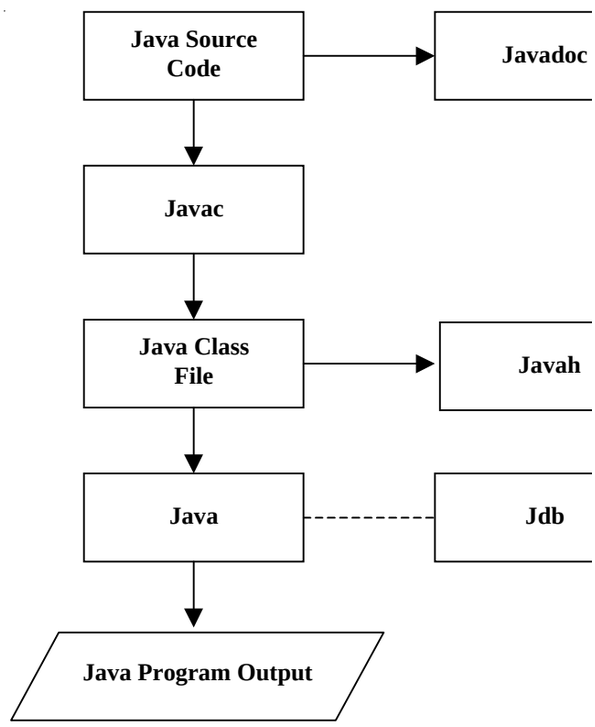
Java environment includes a large number of development tools and hundreds of classes and methods. The development tools are part of Java Development Kit (JDK) and classes and methods are part of Java Standard Library (JSL).

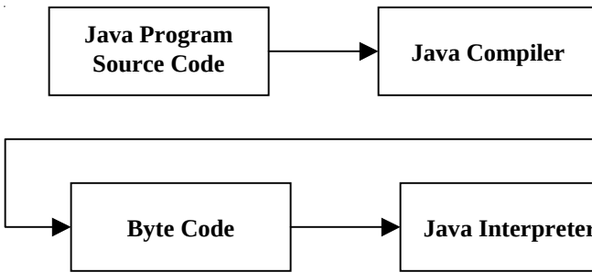
Main Java Development tools are as follows:

Table 15.3: Java Development Tools

Tools	Description
Appletviewer	Enables us to run Java applets (without actually using a Java compatible browser).
Java	Java interpreter, which runs applets and application by reading and interpreting byte codes.
Javac	The java compiler, which translates Java source code to byte code files that the interpreter can understand.
Javadoc	Creates HTML format documentation from Java source code file.
Javah	Produces header file for use with native methods.
Javap	Java disassembler, which enables us to convert byte code files into a program description.
Jdb	Java debugger, which helps us to find errors in our programs.

The way these tools are applied to build and run application program is show below in the figure





Java standard library includes hundred of classes and methods grouped into six functional packages.

1. **Language Support Package:** Collection of classes and methods required for implementing basic features of Java.
2. **Utilities Package:** A collection of classes to provide utility functions such as time and date.
3. **Input /Output Package:** A collection of classes required for input/output manipulation.
4. **Networking Package:** A collection of classes for communicating with other computer via intranet or Internet.
5. **Awt Package:** The abstract window tool kit package contains classes that implements platform independent graphical user interface (GUI).
6. **Applet package:** This includes a set of classes that allow us to create java applets. Applets are small java programs developed for intranet/ Internet applications.

An applet located on a distant computer (server) can be downloaded via internet/ intranet and executed on a local computer (client) using a java capable browser

Structure of Java program

A java program may contain many classes of which only one class defines a main method. A class contains data members and methods that operate on the data member of the class. A java program may contain the following sections:

Documentation section	←	Suggested
Package Statement	←	Optional
Import Statement	←	Optional
Interface statement	←	Optional
Class definitions	←	Optional
Main method class	←	Essential

All Java source files will have the extension Java, also if a program contains multiple classes, the file name must be the class name of the class contains the main method.

Table 15.4: Build-in data types of JAVA

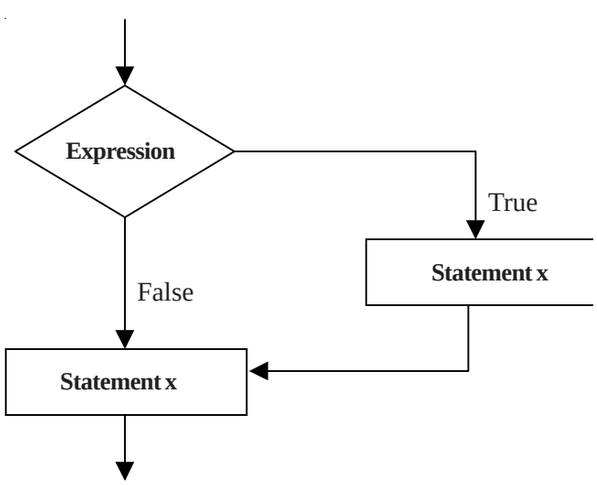
Data Type	Storage Requirement	Range
Byte	1 byte	-128 to 127
Short	2 byte	-32,768 to 32,767
Int	4 bytes	-2,147,483,648 to 2,147,483,647
Long	4 bytes	-9,223,372,036,845,775,808 to 9,223,372,036,845,775,807
Float	4 bytes	3.4e –038 to 3.4e +038
Double	8 byte	1.7e – 308 to 1.7e +308

Control Statements

The if-else statement in java may be implemented in following different form depending on the complexity if condition to be tested:

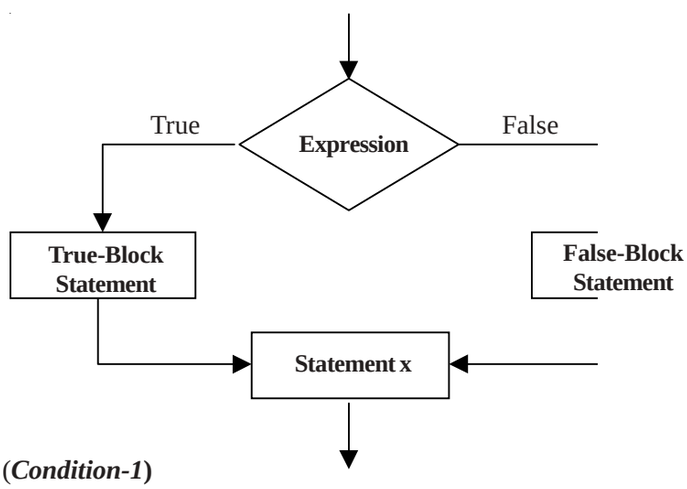
i) Simple if statement
ii) If-else statement
iii) Nested if-else statement
iv) Else if ladder

If (*expression*)
{
Statement(s)
}
Statement x



If (*Expression*)
{
True – block statement(s)
}
else
{
False – block statement(s)
}

Statement – x



If (***Condition-1***)
Statement – 1;
Else if (*Condition – 2)*
Statement – 2;
Else if (*Condition – 3)*
Statement – 3;
.....
.....
.....
Else if (*Condition – n)*
Statement – n;
Else
Default statements;
Statement – x

Activity A

1. What are the different types of conditional branching statement sin Visual basic?

.....

.....

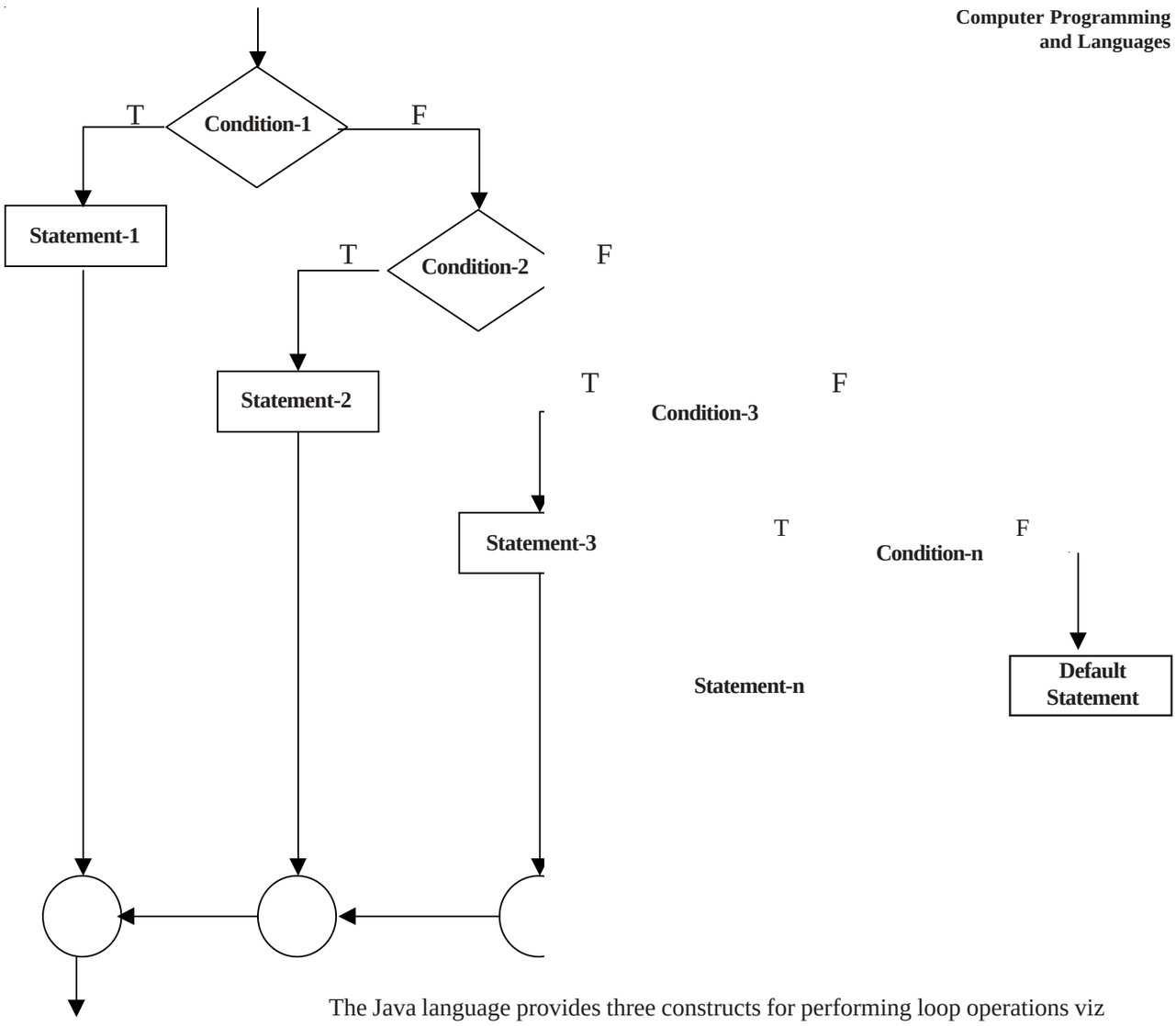
.....

2. Describe the six functional packages in JAVA progam library.

.....

.....

 .....



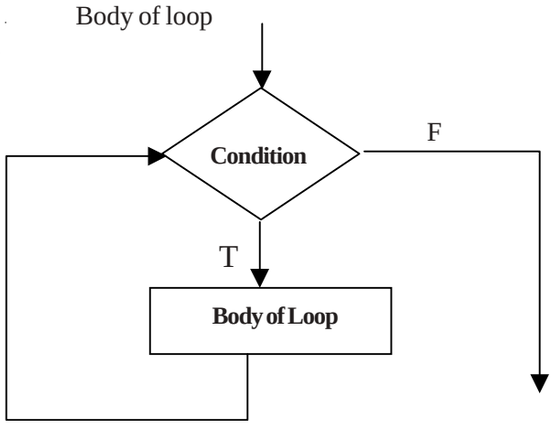
The Java language provides three constructs for performing loop operations viz

While loop
Do loop
For loop

In the while loop test condition is evaluated and if the condition is true, then body of loop is executed as illustrated in the Figure below:

Syntax:

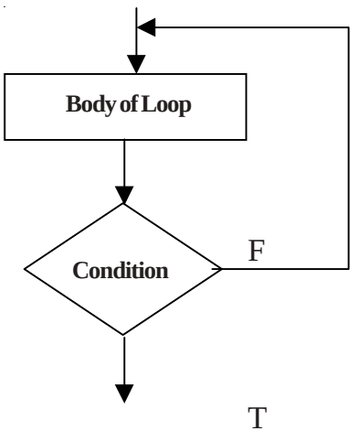
```
Initialization
While (Condition)
    {
        Body of loop
    }
```



In do statement test condition is evaluated the bottom of the loop hence resulting at least one time execution of the body of the loop irrespective of the value of test condition:

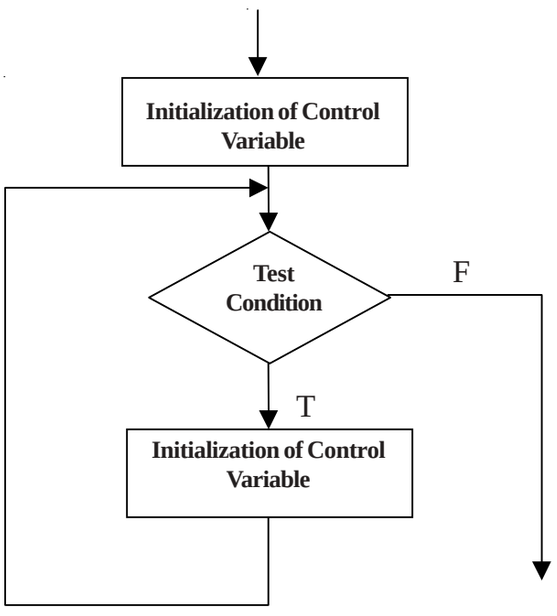
Syntax:

```
Initialization
Do
{
        Body of loop;
} While (condition)
```



The *for* loop is used when we know in advance how many times the loop will run. The syntax is:

```
For (initialization; test condition; increment)
        {
                Body of loop;
        }
```



15. 7 OVERVIEW AND FEATURES OF HYPERTEXT MARKUP LANGUAGE

As a web information provider, we prepare documents in a markup language known as Hypertext Markup Language (HTML). HTML is not a page description language (like Past Script). With HTML, we can describe the structure of our web documents, including related information such as the integration of multimedia and use of hyperlinks. It also links java applets to our web site.

HTML is actually a subset of an internationally known standard called Standard Generalized Markup Language (SGML). The advantage of HTML is that we can create its documents with simple ASCII text editor, which is not always true for document based on SGML. HTML defines documents so that any browser running on any computer can read a display them.

The basic HTML commands that are needed to create web pages fall into following categories:

Structural Command

These identify a file as an HTML document and provide information about the data in the HTML file.

Paragraph formatting Command

These specify paragraph endpoints and heading levels.

Character formatting Command

These allow us to apply various styles to the characters in our documents.

List specification Command

HTML supports several list formats including bulleted numbered and definition list.

Hyper linking command

These allow us to provide information about moving from one document to another.

Assest interaction command

These allow us access to multimedia information. Through these commands, we can display graphical images, access sound and provide digital movies for users.

The body of any HTML document is as under:

```
<HTML>
<HEAD>
<TITLE> This is document title </TITLE>
</HEAD>
<BODY>
        :
        :
        :
        This is document  text
```

<BODY>

</HTML>

HTML file has .HTM or .HTML as their file name extension.

Activity B

A Principal wants to make a web site for the information of parents so that they can get information of their progress of their ward in the school as well about various schemes of the school.

1. Make a diagram for the website.
2. Which language would you prefer for that website.

.....

.....

.....

.....

.....

15.8 OVERVIEW AND FEATURES OF COBOL

COBOL is a compiler language that is commonly used to solve data processing in commercial organization. COBOL is not suitable for scientific purpose programming. COBOL is the abbreviation of Command Business Oriented Language.

COBOL source program is written in Code sheet. In the coding sheet each line has five following zones:

i)	Columns 1-6	:	used for sequence number
ii)	Column 7	:	used for continuation mark an asterisk (*) mean comment
iii)	Column 8-11	:	Margin ‘A’ entries
iv)	Column 12-72	:	Margin ‘B’ entries
v)	Column 73-80	:	User for writing identification of program

There are four division in any COBOL program

IDENTIFICATION DIVISION
ENVIRONMENT DIVISION
DATA DIVISION
PROCEDURE DIVISION

Each division may have several sections. Each section may have several paragraphs. Each paragraph may have several sentences. Each sentence may have several statements. Usually, a statement is terminated by a comma (,) or a period (.).

Division heading, section names paragraphs names, level numbers are usually margin B entries.

Identification Division

This is the first division of every COBOL program. It has no direct effect on the execution of the program.

```
IDENTIFICATION   DIVISION
PROGRAM-ID.              <Program Name>
[ AUTHOR.           <author Name> ]
[ INSTALLATION.     <name of installation>]
[ DATE-WRITTEN.     <date>]
[ DATE-COMPILED     <date>]
[ SECURITY.         <status of security>]
[ REMARKS.          <remarks or comments>]
```

The words enclosed with < > are to be understood as generic terms referencing the COBOL language items. The word enclosed within [] are optional and those enclosed with { } are understood to be alternatives.

In identification division the paragraph PROGRAM-ID is essential. The other paragraphs are optional and machine dependent.

Environment Division

This is the second division of a COBOL program and is mostly machine oriented. It contains information regarding the equipments to be used for processing the program and has two section viz. Configuration Section and INPUT-OUTPUT Section. The structure of this division is as under:

```
ENVIRONMENT   DIVISION.
CONFIGURATION   SECTION.
SOURCE-COMPUTER.   <source computer name>
OBJECT   COMPUTER  <object computer entry>
[SPECIAL-NAMES.      <special name entry>]
INPUT-OUTPUT   SECTION.
FILE-CONTROL.              <file control clauses>
[I-O  CONTROL.       <input-output control entries>.
```

Data Division

This division of a COBOL program is used to describe every data item to be processed by the program instructions in the PROCEDURE DIVISION. The input output files, which are named in the Input-Output Section or the Environment Division, are described in detail in the File Section of the Data Division. Another section in this division is the WORKING STORAGE SECTION in which all fields of the data item, which are required for internal processing but are not part of input output, are described.

```
DATA   DIVISION.

FILE   SECTION.

FD <file name>;

RECORDING   MODE IS <mode>
```

[BLOCK CONTAINS [<integer> TO] <integer> { RECORDS / CHARACTERS] }

[; LABEL { RECORD is } {Standard}
{RECORDS are} {Omitted}

[;RECORD CONTAINS [<integer> TO] <integer> CHARACTERS]

[; VALUE OF <data-name IS {data-name} {literal}

[,<data-name> IS {data-name {literal}]

[; DATA {record IS} <data-name> [‘,<data-name>….]

Working Storage Section

Procedure Division

This division contains the actual instruction for manipulation and processing of data with the help of executable statements. This division is usually organized into several paragraphs. A paragraph start with a paragraph name, followed by a period and one or more sentences. A COBOL instruction is a valid combination of verbs and phrases. There are three categories of COBOL statements:

a) Imperative statements, which always start with a verb *e.g*. ADD, MOVE, READ, WRITE etc.

b) Conditional Statement, which direct the computer to find the truth value of a condition and take subsequent branching to one of the alternative paths depending upon the truth value. If statement is one such conditional statement.

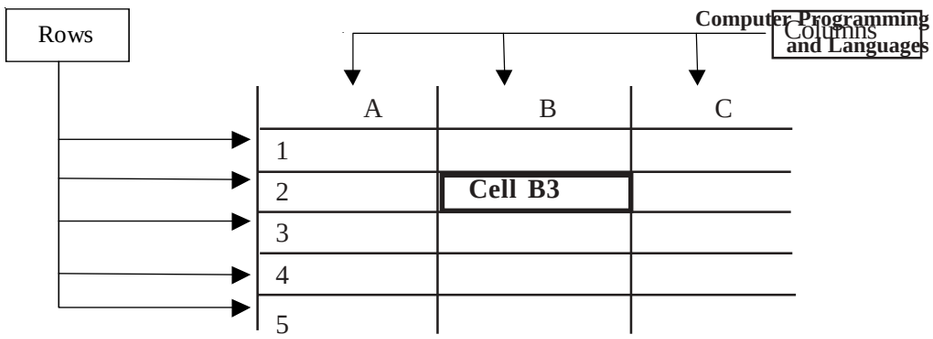
c) Compiler directing statements, which direct the COBOL compiler at the time of compilation and does not generate any instruction in the object program.

15.9 OVERVIEW AND FEATURES OF EXCEL

Microsoft Excel is a program specially designed to enter organized data as well as to analyze and present the data attractively. Excel is one of the most versatile and popular spreadsheet programs. It serves as an, electronic pad for accountants. It an easily perform simple as well as complex mathematical operations. Spreadsheet is a simple worksheet consisting of rows and column in which any data can be entered e.g. report cards of students are manual spreadsheet. An electronic spreadsheet instead of being on papers, it is on the computer screen.

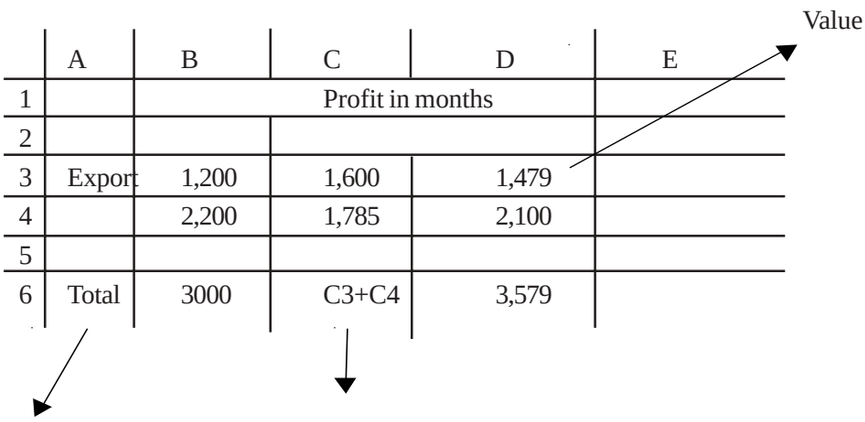
Spreadsheets are used for performing calculations, recalculating results if any data stored in them is changed, creating financial reports, comparing reports etc. A very useful feature of spreadsheet is its ability to create groups. It helps you establish relation-ship between two or more sets of data and easily understand the trends of data changes.

A spreadsheet consists of rows and columns, which combine to form cells. A Cell is a box where we can enter data. Column form the vertical lines of calls while rows form the horizontal line of cells. Cell is an intersection of rows and columns. To describe the location or address of a cell, we have to write the names of the column and the row whose intersection has created this cell.



Cell B3 is location of address of this cell, which means column B, & row 3.

Labels are the headings, which we enter in cell. Values are the number on which calculations are performed. Formulas and functions are elements that perform the desired calculations on the values.



Formulas, which are present within the spreadsheet itself, are called functions. We can also put up our own formulae in any cell. In the above figure cell B6; C6 and D6 contain the formula, which total the values of export and import.

A major drawback of the manual spreadsheet is that if any data in it changes, we have to redo all such calculations that are affected by the data change, but an electronic spreadsheet automatically performs all such calculations for us. All we have to do is just change the data and the rest is taken care of by the spreadsheet.

A spreadsheet can be designed in a number of formats. Basically, designing is concerned with how a number should look on the screen. The various designing formats are comma, percent, currency, scientific. The data in a cell may be right aligned, left aligned or centered.

Exam	**Left-aligned**
Exam	**Right-aligned**
Exam	**Centered**

Spreadsheets allow us to use different fonts and styles to emphasize headings.

Graph is yet another powerful feature of spreadsheet. The data stored in the spreadsheets can be converted into graph so as to study the relationship various data. The spreadsheet program has the facility to draw various graph viz. Bar, Pie, 2D and 2 D graphs.

15.9 SUMMARY

In this unit we have explained the concept and fundamental of programming languages. We have also explained the different type of data types and looping statements available in Visual Basic and Java programming language. We also learn how to compile and run programs in above programming languages.

15.10 UNIT END EXERCISES

1. Explain the concept of programming language. Explain identifiers, constants, expressions, and library functions.
2. Give points to support that Visual Basic is an excellent development tool. Highlight important features of VB.
3. Java is not 100% pure object oriented language? Do you agree or not? Justify your answer.
4. What do you understand by HTML? Present the features of HTML.
5. “Excel is a versatile spreadsheet package. It can do wonder for accountants”. Comment.
6. “With so many ready made and customized software available-The need for a manager is to learn to use them effectively rather than learn to program them.”. Do you agree?
7. What makes Java an almost perfect web programming language? What are its disadvantages?

15.11 REFERENCES AND SUGGESTED FURTHER READINGS

Sebesta, *Concepts of Programming Languages, 4e,* Pearson Education.

Terrence W. Pratt, Marvin V. Zelkowitz , *Programming Languages: Design And Implementation, 4th Ed.,* Prentice-Hall of India.

Tucker, Allen, *Programming Languages Principles and Paradigms,* Tata Mc Graw-Hill